**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ**

**ГОМЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ**

**УНИВЕРСИТЕТ ИМЕНИ П. О. СУХОГО**

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ 4**

по дисциплине:

«Визуальные средства разработки программных приложений»

на тему:

«Коллекции»

Выполнил: студент гр. ИТП-31

Расшивалов Н.И.

Принял: ассистент

Михалевич В.Г.

Гомель 2021

**Цель работы:** изучить работу с коллекциями в *Java*.

**Задание:** Вариант 8. Условия задания на рисунке 1.

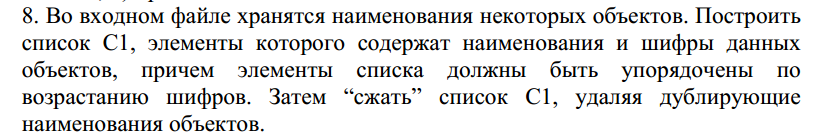


Рисунок 1 – Вариант задания

Структура созданных классов на рисунке 2.

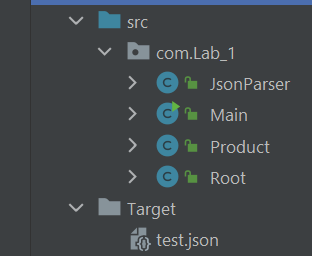


Рисунок 2 – Структура созданных классов

Для выполнения задания был создан файл *test.json,* содержащий определенную информацию о продуктах их шифре.



Рисунок 3 – Фрагмент содержимого файла

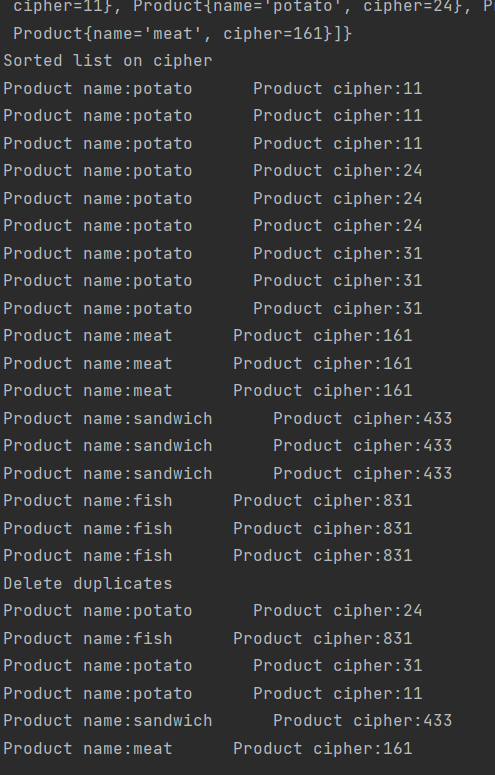


Рисунок 4 – Результат выполнения задания

**Вывод:** Были написаны классы для работы с файлами формата *JSON* и изучена работа с коллекциями в *Java*.

**ПРИЛОЖЕНИЕ А**

Листинг созданных классов

package com.Lab\_1;  
  
import org.json.simple.JSONArray;  
import org.json.simple.JSONObject;  
import org.json.simple.parser.JSONParser;  
import java.io.FileReader;  
import java.util.ArrayList;  
import java.util.List;  
  
public class JsonParser {  
 public Root parse(){  
 Root root = new Root();  
 JSONParser parser = new JSONParser();  
 try( FileReader reader = new FileReader("Target/test.json")) {  
 JSONObject rootJsonObject = (JSONObject) parser.parse(reader);  
 String name = (String) rootJsonObject.get("name");  
 JSONArray productsJsonArray = (JSONArray) rootJsonObject.get("products");  
 List<Product> productList = new ArrayList<>();  
 for (Object item:productsJsonArray){  
 JSONObject productJsonObject = (JSONObject) item;  
 String nameProduct = (String) productJsonObject.get("name");  
 long cipherProduct = (long) productJsonObject.get("cipher");  
 Product product = new Product(nameProduct,(int)cipherProduct);  
 productList.add(product);  
 }  
 root.setName(name);  
 root.setProducts(productList);  
 return root;  
  
 } catch (Exception e) {  
 System.out.println("Parsing error" + e);  
 }  
 return null;  
 }  
}

package com.Lab\_1;  
  
import java.util.Objects;  
  
public class Product {  
 String name;  
 int cipher;  
  
 public Product(String name, int cipher) {  
 this.name = name;  
 this.cipher = cipher;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public int getCipher() {  
 return cipher;  
 }  
  
 public void setCipher(int cipher) {  
 this.cipher = cipher;  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Product product = (Product) o;  
 return cipher == product.cipher && Objects.equals(name, product.name);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.hash(name, cipher);  
 }  
  
 @Override  
 public String toString() {  
 return "Product{" +  
 "name='" + name + '\'' +  
 ", cipher=" + cipher +  
 '}';  
 }  
}

package com.Lab\_1;  
  
import java.util.List;  
import java.util.Objects;  
  
public class Root {  
 String name;  
 List<Product> products;  
  
 public String getName(){return name;}  
  
 public List<Product> getProducts() {return products;}  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 public void setProducts(List<Product> products) {  
 this.products = products;  
 }  
  
 @Override  
 public boolean equals(Object o) {  
 if (this == o) return true;  
 if (o == null || getClass() != o.getClass()) return false;  
 Root root = (Root) o;  
 return Objects.equals(name, root.name) && Objects.equals(products, root.products);  
 }  
  
 @Override  
 public int hashCode() {  
 return Objects.hash(name, products);  
 }  
  
 @Override  
 public String toString() {  
 return "Root{" +  
 "name='" + name + '\'' +  
 ", products=" + products +  
 '}';  
 }  
}